Some Variations on Total Variation–Based Image Smoothing

Antonin Chambolle, Stacey Levine, Bradley Lucier

Some Variations on Total Variation–Based Image Smoothing

Antonin Chambolle,[1] Stacey E. Levine,[2] and Bradley J. Lucier[3]

Abstract

In this paper we study finite-difference approximations to the variational problem using the BV smoothness penalty that was introduced in an image smoothing context by Rudin, Osher, and Fatemi. We give a dual formulation for an "upwind" finite-difference approximation for the BV seminorm; this formulation is in the same spirit as one popularized by Chambolle for a simpler, more anisotropic, finite-difference approximation to the BV seminorm. We introduce a multiscale method for speeding the approximation of both Chambolle's original method and of the new formulation of the upwind scheme. We demonstrate numerically that the multiscale method is effective, and we provide numerical examples that illustrate both the qualitative and quantitative behavior of the solutions of the numerical formulations.

1. Introduction

In an influential paper, Rudin, Osher, and Fatemi [23] suggested using the bounded variation seminorm to smooth images. The functional proposed in their work has since found use in a wide array of problems (see, e.g., [7]), both in image processing and other applications. The novelty of the work was to introduce a method that preserves discontinuities while removing noise and other artifacts. We begin by giving some background on that work.

We work on the unit square $I = [0,1]^2$, where the bounded variation seminorm is defined as

$$\begin{aligned} |f|_{\mathrm{BV}(I)} &:= \int_I |Df(x)|\,dx \\ &:= \sup\Big\{ \int_I f\,\nabla\cdot p \;\Big|\; p : I \to \mathbb{R}^2, \\ &\qquad p \in C^1(I),\ |p(x)| \le 1 \text{ for all } x \in I \Big\} \end{aligned} \tag{1}$$

The formulation of $\mathrm{BV}(I)$ smoothing on which we depend is as follows: Given a nonconstant function f and a number $\lambda > 0$, find the function $\tilde f$ that minimizes over all g

$$\frac{1}{2}\|f-g\|^2_{L_2(I)} + \lambda |g|_{\mathrm{BV}(I)}. \tag{2}$$

Fix f; for each λ this problem has a unique solution that satisfies

$$\frac{\tilde f - f}{\lambda} + \partial_X |\tilde f|_{\mathrm{BV}(I)} \ni 0, \tag{3}$$

where $\partial_X \varphi(g)$ is the subdifferential of the convex, lower-semi-continuous map $\varphi\colon L_2(I) \to \mathbb{R}$. (See [17] for definitions and the basic results we quote here.) Furthermore, if we set

$$\|\tilde f - f\|^2_{L_2(I)} = \sigma^2,$$

then σ^2 is a continuous, one-to-one, increasing function of λ and

$$\sigma^2 < \Big\| f - \int_I f \Big\|^2_{L_2(I)}. \tag{4}$$

Thus, given σ^2 that satisfies (4) there is a unique λ such that the solution $\tilde f$ of (2) given above is also the solution of the problem: Find $\tilde f$ that minimizes over all g with

$$\|f-g\|^2_{\mathrm{BV}(I)} = \sigma^2$$

the functional

$$|g|_{\mathrm{BV}(I)}.$$

Rudin, Osher, and Fatemi introduced $\mathrm{BV}(I)$ image smoothing in the latter form, but these two formulations are equivalent [11].

More recently, the BV smoothing technique is used as just one step in the so-called inverse scale space approach [20, 8].

In the continuous setting, the behavior of the solutions of (2) is well understood (see [22] and references there-in). The qualitative properties of solutions of discrete versions of (2) are not, perhaps, as well known or understood. In this work we study the behavior of solutions of the discrete approach featured in [9] as well as an "upwind" variant of this model that better preserves edges and "isotropic" features.

The paper is organized as follows. In Section 2, we discuss the algorithm introduced in [9] for minimizing a finite-difference approximation to the ROF functional. In Section 3, we propose a new formulation of an upwind finite-difference approximation to the bounded variation seminorm, and show how this can be used to minimize the ROF functional in a manner similar to [9]. In Section 4, we introduce a multiscale algorithm that greatly reduces the run time of both methods. Section 5 contains numerical examples that demonstrate the qualitative properties of the algorithm and observed rates of convergence for two special problems with known solutions.

---

[1] CMAP, Ecole Polytechnique, CNRS, 91128 Palaiseau, France. The work of this author was partially supported by the ANR "MICA" project, grant ANR-08-BLAN-0082.

[2] Department of Mathematics and Computer Science, Duquesne University, 440 College Hall, Pittsburgh, PA 15282, USA. The work of this author was partially supported by NSF-DMS grant #0505729 and the Institute for Mathematics and its Applications, the University of Minnesota.

[3] Department of Mathematics, Purdue University, 150 N. University St., West Lafayette, IN 47907, USA. The work of this author was partially supported by the Office of Naval Research, Contract N00014-91-J-1152, and the Institute for Mathematics and its Applications, the University of Minnesota.

2. Discrete BV(I) Variational Smoothing.

To begin we consider discretizations of (2). A standard approach, as first suggested in [23], is to regularize the BV seminorm and consider the problem of minimizing, with $\epsilon > 0$,

$$\frac{1}{2}\|f-g\|^2_{L_2(I)} + \lambda \int_I \sqrt{|\nabla g|^2 + \epsilon^2}.$$

This functional is differentiable in g, and one can follow the flow of its associated Euler-Lagrange equation; numerical methods approximate this flow. This is sufficient for some applications, but the solution now depends on the regularization parameter ϵ.

In this work we consider a discrete analog of (2) and follow the dual approach of in [9]. The material in this section is classical; we refer the reader to [9] for a more extended treatment, and to [13] which puts [9] into some historical context. Another approach to solving the discrete problem that arises here is through second-order cone programming [18], and through graph cuts [14], where one considers anisotropic approximations to the usual isotropic BV seminorm.

Given $N > 1$, we let $h = 1/N$ and consider discrete functions

$$f_i, \quad i = (i_1, i_2),\ 0 \le i_1, i_2 < N.$$

A discrete $L_2^h(I)$ norm of f is defined by

$$\|f\|^2_{L_2^h(I)} = \sum_{0 \le i_1, i_2 < N} |f_i|^2\, h^2.$$

One way to compute a discrete gradient of a discrete scalar function f_i is given by

$$\nabla_h f_i := \Big(\frac{f_{i+(1,0)} - f_i}{h}, \frac{f_{i+(0,1)} - f_i}{h}\Big). \tag{5}$$

For any discrete gradient operator, one needs to specify the value of f_i for some values of i outside $[0,N)^2$; to do so, we need to specify boundary conditions. We assume that scalar discrete functions g, f, etc., are either periodic (with period N) or satisfy Neumann or Dirichlet boundary conditions. For Dirichlet boundary conditions, f_i is zero for i outside $[0,N)^2$; for Neumann conditions, we consider f to be reflected across the lines $i_1 = N - 1/2$ and $i_2 = N - 1/2$ and then extended periodically across the plane with period $2N$.

Given a discrete gradient, we can define an associated discrete BV seminorm

$$|f|_{\mathrm{BV}^h(I)} = \sum_{0 \le i_1, i_2 < N} |\nabla_h f_i|\, h^2$$

and then in turn a discrete analogue to (2),

$$\frac{1}{2}\|f-g\|^2_{L_2^h(I)} + \lambda |g|_{\mathrm{BV}^h(I)}. \tag{6}$$

For any discrete gradient, we define an associated discrete divergence of vector functions $p_i = (p_i^{(1)}, p_i^{(2)})$; given the discrete gradient (5), we define the associated divergence by

$$\nabla_h \cdot p_i = \frac{p_i^{(1)} - p_{i-(1,0)}^{(1)}}{h} + \frac{p_i^{(2)} - p_{i-(0,1)}^{(2)}}{h}. \tag{7}$$

As for boundary conditions, we note that in the following we compute discrete divergences only of discrete vector fields that are themselves discrete gradients; therefore we compute p_i for i outside of $[0,N)^2$ in a manner consistent with whatever boundary condition we have chosen for discrete scalar functions f_i.

Because of how the discrete gradient and divergence are related, we have

$$\begin{aligned}
|f|_{\mathrm{BV}^h(I)} &= \sum_{0 \le i_1, i_2 < N} |\nabla_h f_i|\, h^2 \\
&= \sup_{|p_i| \le 1} \sum_{0 \le i_1, i_2 < N} (-\nabla_h) f_i \cdot p_i\, h^2 \\
&= \sup_{|p_i| \le 1} \sum_{0 \le i_1, i_2 < N} f_i \big(\nabla_h \cdot p_i\big)\, h^2.
\end{aligned} \tag{8}$$

The first equality is obvious; the second (which can be interpreted as "the adjoint of the discrete divergence is the negative of the discrete gradient") follows by summation by parts.

Thus, if the symmetric convex set $\overline{K}$ is defined by

$$\overline{K} = \{g_i = \nabla_h \cdot p_i \mid |p_i| \le 1,\ p_i = (p_i^{(1)}, p_i^{(2)})\},$$

then

$$|f|_{\mathrm{BV}^h(I)} = \sup_{g \in \overline{K}} \sum f_i g_i\, h^2 =: \langle f, g\rangle.$$

Using classical convexity arguments, the first author [9] showed that the minimizer $\tilde f$ of (6) is $f - \lambda \nabla_h \cdot \bar p$ with $\bar p$ a minimizer of

$$F(p) := \Big\|\nabla_h \cdot p - \frac{f}{\lambda}\Big\|^2 \tag{9}$$

subject to the constraint

$$p \in K := \{p \colon [0,N)^2 \to \mathbb{R}^2 \mid |p_i| \le 1 \text{ for all } i\}. \tag{10}$$

In other words,

$$\tilde f = f - \pi_{\lambda \overline{K}} f, \tag{11}$$

where $\pi_{\lambda\overline{K}}$ is the orthogonal projector in $L_2^h(I)$ of f onto the convex set $\lambda\overline{K}$.

Chambolle [9] gave a specific iterative algorithm for finding a discrete vector field p that minimizes (11). He first set $p_i^0 = 0$ for all i (so that p^0 is obviously in K) and then calculates

$$p_i^{n+1} := \frac{p_i^n - \tau(-\nabla_h)(\nabla_h \cdot p^n - f/\lambda)_i}{1 + \tau|(-\nabla_h)(\nabla_h \cdot p^n - f/\lambda)_i|}. \tag{12}$$

He notes that $p^n \in K$ for all n and shows, using the Karush-Kuhn-Tucker theorem, that if $0 < \tau \le h^2/8$ the limit $\lim_{n\to\infty} p^n$ exists and gives a minimizer of (11) over all p with $|p_i| \le 1$. Procedure (12) can be written as the two-step process

$$p_i^{n+1/2} := p_i^n - \tau(-\nabla_h)(\nabla_h \cdot p^n - f/\lambda)_i,$$
$$p_i^{n+1} := \frac{p^{n+1/2}}{1 + |p_i^{n+1/2} - p_i^n|},$$

where the first formula is just gradient descent of the functional $F(p)$ with step τ, and the second is a nonlinear projector that ensures that $p^{n+1} \in K$ if $p_i^n \in K$.

In [10], Chambolle speculates whether the two-step procedure

$$\begin{aligned} p_i^{n+1/2} &:= p_i^n - \tau(-\nabla_h)(\nabla_h \cdot p^n - f/\lambda)_i, \\ p_i^{n+1} &:= \frac{p_i^{n+1/2}}{\max(1, |p_i^{n+1/2}|)} \end{aligned} \tag{13}$$

yields vector fields p^n such that $\nabla_h \cdot p^n$ converges to the projection $\pi_{\overline{K}}(f/\lambda)$. Again, the first half-step is gradient descent along the functional $F(p)$ given in (23), while the second is the L_2^h-projection of p onto the set K. We note at the end of the next section that this iteration is a special case of a more general minimization algorithm developed originally by Eicke [16], which itself has been generalized by Combettes [12, 13]. There is recent work related to algorithms of this type by Aujol [4].

3. Upwind BV Smoothing

One might think that, given $g \in \mathrm{BV}(I)$, the $L_2(I)$ projection of g on a grid with sidelength h

$$g_i = \frac{1}{h^2}\int_{I_i} g, \quad I_i = h(I+i) \tag{14}$$

or the (multi-valued) $L_1(I)$ projection of g on the same grid

$$\begin{aligned} g_i = \text{any } m \text{ such that } &|\{x \in I_i \mid g(x) \ge m\}| \ge 1/2 \\ \text{and } &|\{x \in I_i \mid g(x) \le m\}| \ge 1/2 \end{aligned}$$

would satisfy

$$\lim_{h\to 0} |g|_{\mathrm{BV}^h(I)} = |g|_{\mathrm{BV}(I)}, \tag{15}$$

but this is not true in general. If g is C^1, then integration by parts in (1) shows that (15) holds. Again, if g is the characteristic function of the set $\{x_1 < 1/2\}$ or $\{x_2 < 1/2\}$, or even $\{x_1 + x_2 < 1\}$, then (15) holds, but if g is the characteristic function of $\{x_1 < x_2\}$ and we use the projection (14) then a calculation shows that

$$2 = \lim_{h\to 0} |g|_{\mathrm{BV}^h(I)} \ne |g|_{\mathrm{BV}(I)} = \sqrt{2}\,. \tag{16}$$

In fact, this is not so much of an issue as far as *minimization problems* are concerned, since it is well known in this case that the correct notion of convergence is Γ-*convergence* [6], which ensures convergence of the minimizers of variational problems. It can be shown without much difficulty that the semi-norms $|\cdot|_{BV^h}$ Γ-converge as $h \to 0$ to the BV seminorm. However, it follows from the inequality (16) that the approximation of $\chi_{\{x_1<x_2\}}$, as a possible solution of a minimization problem, will be possible only after some smoothing of the discontinuity, so that the output of a discrete minimization will usually not be as sharp as one could hope.

This issue motivates us to we define an "upwind" discrete $\mathrm{BV}(I)$ norm of a discrete scalar function g_i given by

$$\|g\|_{\mathrm{BV}^h(I)} := \sum_i |(-\nabla_h)g_i \vee 0|\, h^2, \tag{17}$$

where we have defined the discrete gradient

$$(-\nabla_h)g_i = \begin{pmatrix} \dfrac{g_i - g_{i+(1,0)}}{h} \\ \dfrac{g_i - g_{i-(1,0)}}{h} \\ \dfrac{g_i - g_{i+(0,1)}}{h} \\ \dfrac{g_i - g_{i-(0,1)}}{h} \end{pmatrix}. \tag{18}$$

and we denote by $p \vee q$ and $p \wedge q$ the componentwise maximum and minimum, respectively, of the vectors p and q. (Similarly, if we write an inequality between vectors, $p \le q$, then we mean that this inequality holds componentwise.) This type of operator is based on the classical first-order upwind finite-difference scheme used to solve hyperbolic partial differential equations; upwind methods have found important applications in level set methods [21].

Of course, in the present case, the problem we are solving is degenerate elliptic and there is, strictly speaking, no direction of "wind" so that our choice may be seen as quite arbitrary (and, in fact, the reverse direction is also an admissible choice). However it still produces the desired effect, which is to preserve some discontinuities better than standard discretizations. We may refer, for a similar idea, to Appleton and Talbot [3] who recently proposed to compute minimal surfaces by solving some hyperbolic system, discretized with an upwind scheme. Also in their case the "speed" and "wind" can be reversed, still, their approach produces sharp discontinuities as desired — on the other hand, it does not really correspond to the minimization of a convex discrete functional such as our upwind TV.

In (17) we include a difference in the sum only if the difference is *positive*, i.e., the discrete function g_i is *increasing* as it goes to g_i from the given direction. Note that for smooth $g(x)$ this is a convergent approximation to the BV seminorm of f and for jumps across vertical, horizontal, or diagonal lines you get the correct value of the BV semi-norm; that is, $\lim_{h\to 0} |g|_{\mathrm{BV}^h(I)} = |g|_{\mathrm{BV}(I)}$.

We then write this "upwind" semi-norm as

$$\sum_i |(-\nabla_h)g_i \vee 0|\, h^2 = \sup_{|p_i|\le 1,\ p_i \ge 0} \sum_i (-\nabla_h)g_i \cdot p_i\, h^2.$$

Here we require not only that the Euclidean norm $|p_i|$ of $p_i \in \mathbb{R}^4$ be no larger than one, but also that each coordinate of p_i be non-negative, so that p is in the set

$$K := \{p\colon [0,N)^2 \to \mathbb{R}^4 \mid |p_i| \le 1 \text{ and } p_i \ge 0\}. \tag{19}$$

Thus we have dealt with the "extra" nonlinearity of (17) by incorporating it into the convex set that contains p.

If we now define the discrete divergence that is the adjoint of the discrete gradient (18),

$$\begin{aligned}(20)\qquad \nabla_h\cdot\xi_i = {} & \frac{\xi_i^{(1)}-\xi_{i-(1,0)}^{(1)}}{h}+\frac{\xi_i^{(2)}-\xi_{i+(1,0)}^{(2)}}{h}\\ & +\frac{\xi_i^{(3)}-\xi_{i-(0,1)}^{(3)}}{h}+\frac{\xi_i^{(4)}-\xi_{i+(0,1)}^{(4)}}{h}\end{aligned}$$

and again apply summation by parts, we see that this new discrete semi-norm is equal to

$$\sup_u \sum_i g_i u_i\, h^2 = \sup_u \langle g,u\rangle$$

with u in the convex set

$$(21)\quad \overline{K} := \{\nabla_h\cdot p \mid p:[0,N)^2\to\mathbb{R}^4,\ |p_i|\le 1,\ p_i\ge 0\Big\}.$$

Thus, as in [9], the minimizer over discrete scalar functions g_i of

$$(22)\qquad \frac{1}{2}\|f-g\|^2+\lambda\|\,|(-\nabla_h)g\vee 0|\,\|_1$$

can be written as the difference between f and the unique projection of f onto the convex set $\lambda\overline{K}$.

In other words, the minimizer of (22) is $f-\lambda\nabla_h\cdot\bar p$ where $\bar p$ is any minimizer of the functional

$$(23)\qquad F(p) := \Big\|\nabla_h\cdot p-\frac{f}{\lambda}\Big\|^2$$

subject to the constraint that $p\in K$, where K is defined by (19). An iterative method to compute a p that minimizes (23) is given by

$$\begin{aligned}&p_i^{n+1/3} := p_i^n-\tau(-\nabla_h)(\nabla_h\cdot p^n-f/\lambda)_i,\\ (24)\qquad &p_i^{n+2/3} := p_i^{n+1/3}\vee 0,\\ &p_i^{n+1} := \frac{p^{n+2/3}}{\max(1,|p^{n+2/3}|)},\end{aligned}$$

where p_i^0 is chosen arbitrarily in K. The computation of $p_i^{n+1/3}$ is simply gradient descent of (23), while the next two steps compute the projection of $p^{n+1/3}$ onto K.

We remark that [24] contains bounds for the difference in $L_2(I)$ between discrete minimizers of (22) and the minimizer of (2) as the mesh size $h\to 0$.

The notational similarity of (23) and (11) is deliberate; both can be formulated as minimizing over all x in a closed convex set K

$$(25)\qquad F(x)=\|Ax-b\|^2$$

for some bounded linear operator A and vector b. In our cases we have $x=p$, $Ax=\nabla_h\cdot p$, $b=f/\lambda$, and K is given by (10) or (19).

We consider the general iteration

$$(26)\qquad x^{n+1}=\pi_K(x^n-\tau(A^*(Ax^n-b))),$$

where π_K is the orthogonal projection onto the set K, A^* is the adjoint of the operator A, and τ is suitably small. In other words, we first perform gradient descent on the functional $F(x)$ and then project the intermediate result onto the convex set K.

The convergence of this algorithm was studied by Eicke [16], Theorem 3.2, and is a special case of a general theory developed later by Combettes and his collaborators [12, 13]. For pedagogical purposes we recommend the analysis in [16], which is particularly short and self contained. The result applied to our (finite-dimensional) problem gives the following: if $0<\tau<2\|A\|^{-2}$ then x^n converges to a minimizer of $F(x)$ on K. (Part of the result goes back to Opial [19].) In our case this means that the method converges if $0<\tau<h^2/8$ for the discrete divergence (7) [9]; a similar argument shows that for the discrete divergence (20) we obtain convergence when $0<\tau<h^2/16$.

The iteration (26) is efficient only if π_K, A, and A^* can be calculated quickly. In our case, it takes $O(N^2)$ operations to calculate $\nabla_n\cdot p$ or $-\nabla_h f$ on an $N\times N$ image. For the set K defined by (10), we have simply

$$(\pi_K p)_i=\frac{p_i}{\max(1,|p_i|)}.$$

For K defined by (19) we set

$$(\pi_K p)_i=\frac{\bar p_i}{\max(1,|\bar p_i|)}\ \text{where } \bar p_i=p_i\vee 0.$$

So with either (10) or (19) we can calculate $\pi_K p$ on an $N\times N$ image in $O(N^2)$ operations.

4. A Multiscale Algorithm

The characterization of the minimizer (11) allows us the following observation: We need only construct a vector field p that minimizes $F(p)$ over all $p\in K$. Any general iteration of the form (26) converges as long as the initial data is in the set K. We propose here to use a multiscale technique to get a good approximation $p^0\in K$ for our iterations.

We consider two grids in I, one with grid spacing $2h$ and one with grid spacing h. We will construct a scalar injector from the $2h$-grid to the h-grid (called I_{2h}^h) and a scalar projector from the h grid to the $2h$ grid (called I_h^{2h}). Similarly, we will have an operator-dependent injector $\vec{I}_{2h}^h$ on vector fields. Our general approach will then be as follows.

Given data f_h on a grid with spacing h, we calculate data

$$f_{2h}=I_h^{2h}f_h$$

on a grid with spacing $2h$. We then calculate the minimizer p_{2h} of (23) with data f_{2h} using our iterative algorithm (not yet specifying the initial value p^0). Next, we begin the iteration solving (23) with data f_h with the initial vector field

$$p^0=\vec{I}_{2h}^h p_{2h}.$$

We now explain our choice of I_h^{2h} and $\vec{I}_{2h}^h$. Assume that N is even with $h = 1/N$, and define the discrete inner product on $N \times N$ arrays

$$\langle u, v\rangle_h = \sum_i u_i v_i \, h^2$$

where the sum is taken over all $i = (i_1, i_2)$ with $0 \le i_1, i_2 < N$. A similar inner product can be defined on $N/2 \times N/2$ arrays with grid spacing $2h$:

$$\langle u, v\rangle_{2h} = \sum_i u_i v_i \, (2h)^2,$$

where the sum now is over all $i = (i_1, i_2)$ with $0 \le i_1, i_2 < N/2$.

Our injector I_{2h}^h will simply be the constant injector on 2×2 squares:

$$(I_{2h}^h u)_i = u_{\lfloor i/2 \rfloor},$$

where $\lfloor i/2 \rfloor = (\lfloor i_1/2 \rfloor, \lfloor i_2/2 \rfloor)$ and $\lfloor x \rfloor$ is the largest integer no greater than x. The corresponding projector I_h^{2h} is defined as the adjoint of I_{2h}^h with respect to the h- and $2h$-inner products, i.e., for an $N \times N$ grid function v and a $N/2 \times N/2$ grid function u

$$\langle I_{2h}^h u, v\rangle_h = \langle u, I_h^{2h} v\rangle_{2h}.$$

A direct calculation shows that

$$(I_h^{2h} v)_i = \frac{1}{4}\big(v_{2i} + v_{2i+(1,0)} + v_{2i+(0,1)} + v_{2i+(1,1)}\big),$$

i.e., it is simply the average of the values of v on 2×2 subgrids. (The factor 1/4 comes in because of the different weights in the two inner products.)

After we calculate the minimizer p_{2h} of (23) over all $p \in K$ on the grid with spacing $2h$, we start the iteration on the grid with spacing h with

$$p^0 = \vec{I}_{2h}^h p_{2h}.$$

Here $\vec{I}_{h2}^h$ is an injector that satisfies

$$\nabla_h \cdot \vec{I}_{2h}^h p_{2h} = I_{2h}^h \nabla_{2h} \cdot p_{2h}. \tag{27}$$

Specifically, for the anisotropic operators (7) and (5) we have

$$(\vec{I}_{2h}^h p)_i = \frac{1}{2}\begin{pmatrix} p^{(1)}_{\lfloor i/2 \rfloor} + p^{(1)}_{\lfloor (i-(1,0))/2 \rfloor} \\ p^{(2)}_{\lfloor i/2 \rfloor} + p^{(2)}_{\lfloor (i-(0,1))/2 \rfloor} \end{pmatrix},$$

and for the upwind operators (18) and (20) we have

$$(\vec{I}_{2h}^h p)_i = \frac{1}{2}\begin{pmatrix} p^{(1)}_{\lfloor i/2 \rfloor} + p^{(1)}_{\lfloor (i-(1,0))/2 \rfloor} \\ p^{(2)}_{\lfloor i/2 \rfloor} + p^{(2)}_{\lfloor (i+(1,0))/2 \rfloor} \\ p^{(3)}_{\lfloor i/2 \rfloor} + p^{(3)}_{\lfloor (i-(0,1))/2 \rfloor} \\ p^{(4)}_{\lfloor i/2 \rfloor} + p^{(4)}_{\lfloor (i+(0,1))/2 \rfloor} \end{pmatrix}.$$

In both cases, we know p_{2h} minimizes

$$\|\nabla_{2h} \cdot p - f_{2h}/\lambda\|_{2h}^2$$

over all $p \in K_{2h}$. Because (27) holds we know that $\vec{I}_{2h}^h p_{2h}$ minimizes over all $p \in \vec{I}_{2h}^h K_{2h} \not\subseteq K_h$

$$\|\nabla_h \cdot p - I_{2h}^h f_{2h}/\lambda\|_h^2.$$

We also know that

$$I_{2h}^h f_{2h} - \lambda \nabla_h \cdot \vec{I}_{2h}^h p_{2h}$$

minimizes

$$\frac{1}{2}\|I_{2h}^h f_{2h} - g\|_h^2 + \lambda |g|_{\mathrm{BV}^h}$$

over all $g \in I_{2h}^h \overline{K}_{2h} \not\subseteq \overline{K}_h$.

We don't know a reasonable way to bound the error incurred by starting the iteration on the grid with spacing h with

$$p^0 = \pi_{K_h} \vec{I}_{2h}^h p_{2h}.$$

There are three sources of error: (1) we use $I_{2h}^h f_{2h} = I_{2h}^h I_h^{2h} f_h$ as the data instead of f_h; (2) we minimize over all $p \in \vec{I}_{2h}^h K_{2h}$ instead of over K_h, and (3) we immediately project $\vec{I}_{2h}^h p_{2h}$ onto K_h. It is straightforward to bound the first error, since the solutions to all these problems are L_2 contractions so the difference between the minimizers of

$$\frac{1}{2}\|I_{2h}^h f_{2h} - g\|_h^2 + \lambda |g|_{\mathrm{BV}^h}$$

and

$$\frac{1}{2}\|f_h - g\|_h^2 + \lambda |g|_{\mathrm{BV}^h}$$

over all $g \in I_{2h}^h \bar{K}_{2h}$ is bounded by

$$\begin{aligned}\|f_h - I_{2h}^h f_{2h}\|_{L_2} &\le \|f - f_h\|_{L_2} + \|f - f_{2h}\|_{L_2} \\ &\le C h^\alpha |f|_{\mathrm{Lip}(\alpha, L_2)}\end{aligned}$$

whenever f is in the Lipschitz space $\mathrm{Lip}(\alpha, L_2)$. We don't know how to deal with the other two errors.

5. Experimental Results

We did a series of experiments to (a) measure the effectiveness of the multiscale predictor for the initial vector field p, (b) examine the experimental convergence rates for two sets of initial data for the continuous BV problem with known analytic solutions, and (c) illustrate some of the qualitative properties of the solutions to the discrete problems.

All of our algorithms compute only an approximate minimizer of discrete BV problems—we must decide when to stop the iteration (26). In [10] one finds a simple error bound; first we let

$$\tilde{f}^n = f - \lambda \nabla_h \cdot p^n.$$

Then we have

$$\|\tilde{f} - \tilde{f}^n\|_{L_2^h(I)}^2 \le \lambda(|\tilde{f}^n|_{\mathrm{BV}^h(I)} - \langle \nabla_h f^n, p^n\rangle) = \epsilon(p^n)^2. \tag{28}$$

So we can ensure that we've computed $\tilde{f}$ to within an error of ϵ, i.e.,

$$\|\tilde{f} - \tilde{f}^n\|_{L_2^h(I)} \le \epsilon,$$

if we iterate until $\epsilon(p^n) \le \epsilon$. It is possible that the true error is much less than the bound, but we don't know that.

In our computations the initial data is 255 times the characteristic function of a disk or of a square, and we set $\epsilon(p^n) = 1/4$, so we ensure that the $L_2^h(I)$ error between our computed discrete minimizer and the exact discrete minimizer is $\le 1/4$ greyscales.

We begin by discussing the efficiency improvements we observed by our multiscale method. As mentioned in the previous paragraph, we do not compute exact discrete minimizers, but only approximate discrete minimizers. Our final goal is to compute $\tilde{f}_h^n$ such that

$$\|\tilde{f}_h - \tilde{f}_h^n\|_{L_2^h(I)} \le \epsilon_h$$

for some ϵ_h. We begin the iterative algorithm with some initial guess for p_h^0. Without our multiscale algorithm we take $p_h^0 = 0$. Our multiscale algorithm says that we should iteratively compute p_{2h}^n on a grid with grid size $2h$ until

$$\|\tilde{f}_{2h} - \tilde{f}_{2h}^n\|_{L_2^{2h}(I)} \le \epsilon_{2h}$$

and then set

$$p_h^0 = \pi_{K_h} \vec{I}_{2h}^h p_{2h}^n.$$

In our computations we take $\epsilon_{2h} = \epsilon_h = 1/4$. It is possible that, given $\epsilon_h = 1/4$, a better choice of ϵ_{2h} can improve our initial vector field p_h^0.

We compare the computational effort needed by our multiscale method and the iterative method operating solely on the grid with grid spacing h and $p_h^0 = 0$. We note that each iteration $p_{2h}^n \to p_{2h}^{n+1}$ on a grid with mesh size $2h$ takes (roughly) 1/4 as many operations as one iteration on the grid of size h. Thus, if the number of iterations on grids with spacing h, $2h$, $4h$, etc., are N_h, N_{2h}, N_{4h}, respectively, we report the number of equivalent iterations on the finest grid,

$$N_h + \frac{1}{4}N_{2h} + \frac{1}{16}N_{2h} + \cdots.$$

In our examples we take the data to be $f = 255\chi_{[\frac14,\frac34]^2}$, the characteristic function of a subsquare with sidelength 1/2 inside the computational domain of $[0,1]^2$. We use Dirichlet boundary conditions. We computed numerical solutions on grids with $h = 1/128$, $1/256$, and $1/512$. We chose three values of λ for which the $L_2(I)$ distance between the solution of the continuous problem (2) and the initial data f is 16, 32, and 64. For this purpose we use the characterization of the exact solutions (see [5], Section 4 in [2], or Appendix 1 in [1]) and found that the corresponding values of λ are 3.771636443, 7.820179629, and 16.26268646, respectively.

A summary of the iteration count to solve both problems (6) and (22) such that the error bound (28) is $< 1/4$, both with and without the multiscale approach, are summarized in Tables 1 and 2. The iteration count with the multiscale approach is reported as the equivalent number of iterations at the finest resolutions as described above.

Table 1
Iteration count without the multiscale algorithm; columns 1–3 are the result of (6)*; columns 4–6 are the result of* (22)

	16	32	64	16	32	64
128	4,815	21,772	119,468	4,293	5,414	13,049
256	10,466	45,744	255,096	17,173	21,162	33,158
512	36,653	103,817	514,060	68,324	83,908	113,843

Table 2
Iteration count with the multiscale algorithm; columns 1–3 are the result of (6)*; columns 4–6 are the result of* (22)

	16	32	64	16	32	64
128	1,393	2,358	10,047	1,694	2,574	3,476
256	4,525	6,722	12,250	5,460	8,851	12,484
512	14,615	22,328	33,115	17,197	30,676	44,289

We see that the multiscale method of choosing p_h^0 speeds up the computation, and greatly so in some cases.

Next we discuss the observed error between the minimizer $\tilde{f}$ of the continuous problem (2) and the approximate minimizers $\tilde{f}_h^n$ (with an error in $L_2^h(I)$ of < 0.25) of the two discrete problems (6) and (22). For the three problems with

$$f = 255\chi_{[\frac14,\frac34]^2}, \tag{29}$$

and λ equal to 3.771636443, 7.820179629, and 16.26268646 (corresponding to $\|f - \tilde{f}\|_{L_2(I)} = 16$, 32, and 64, respectively) we computed numerical approximation to the exact solutions on a grid of size 2048×2048; the value of the numerical approximation on the subsquare $\frac{1}{2048}(I+i)$ is taken to be

$$\tilde{f}\Big(\frac{1}{2048}\Big(i + \Big(\frac12,\frac12\Big)\Big)\Big). \tag{30}$$

A simple geometric argument shows that this approximation is a near-best piecewise constant projection in $L_2(I)$ of $\tilde{f}$ onto a 2,048 × 2,048 grid (just follow the argument for Example 2 in Section III.E of [15], as the measure of the subgrid square where $\tilde{f}$ is less than the value (30) is no less than 1/4, and no greater than 3/4, times the measure of the subgrid square).

Table 3
$L_2(I)$ errors on grids of size 128, 256, *and* 512, *and differences $\|f - \tilde{f}\|_{L_2(I)}$ of* 16, 32, *and* 64, *with initial data* (29)*; columns 1–3 are the result of* (6)*; columns 4–6 are the result of* (22)*; α is the estimated order of convergence, $\|\tilde{f} - \tilde{f}_h\|_{L_2(I)} \approx Ch^\alpha$.*

	16	32	64	16	32	64
128	1.613	1.889	2.113	1.533	1.813	2.045
256	0.962	1.134	1.249	0.900	1.041	1.145
512	0.554	0.654	0.733	0.508	0.578	0.639
α	0.772	0.765	0.764	0.796	0.824	0.839

We compute piecewise constant approximations f_h^n of the minimizers of (6) and (22) on grids of size 128×128,

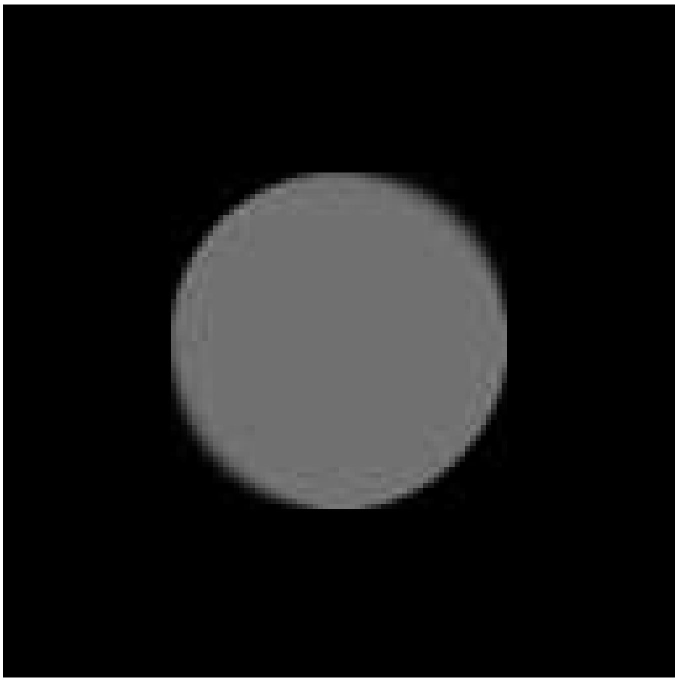

FIGURE 1. *The solution of the discrete BV problem using the anisotropic definition* (8).

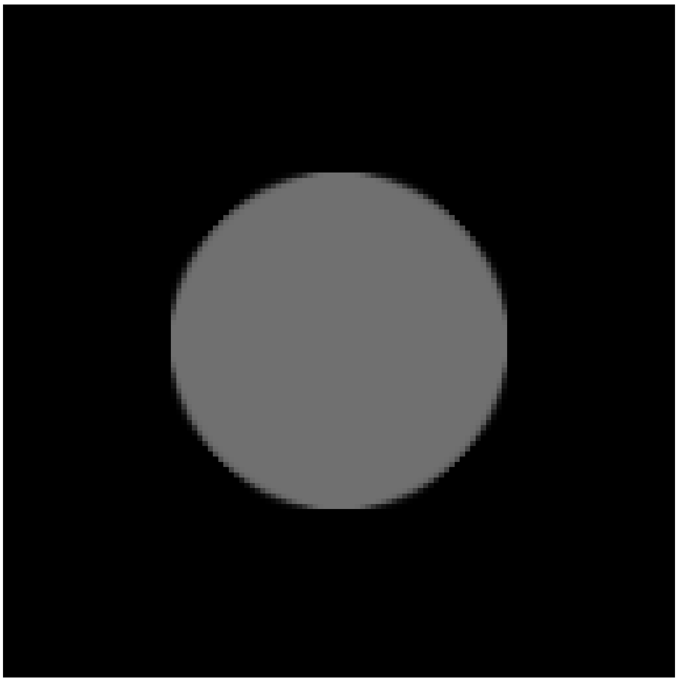

FIGURE 2. *The solution of the discrete BV problem using the upwind definition* (17).

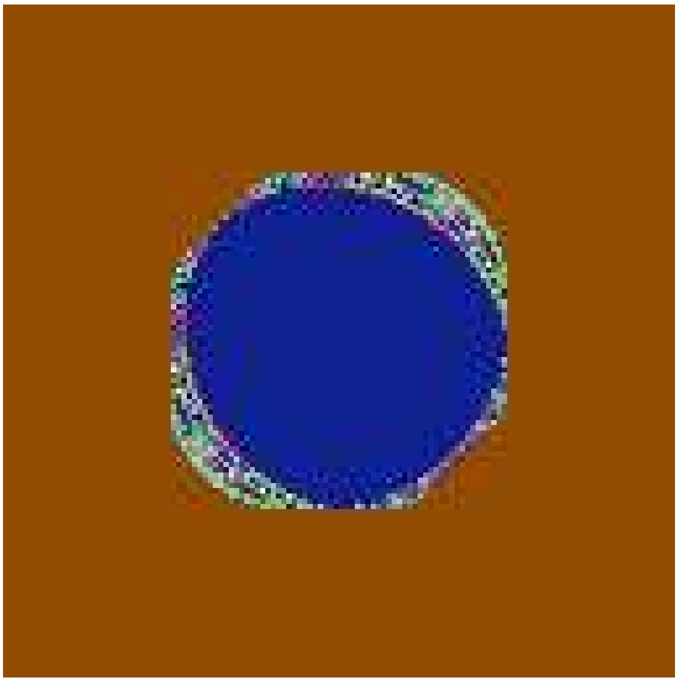

FIGURE 3. *The solution of the discrete BV problem using the anisotropic definition* (8) *in "false color".*

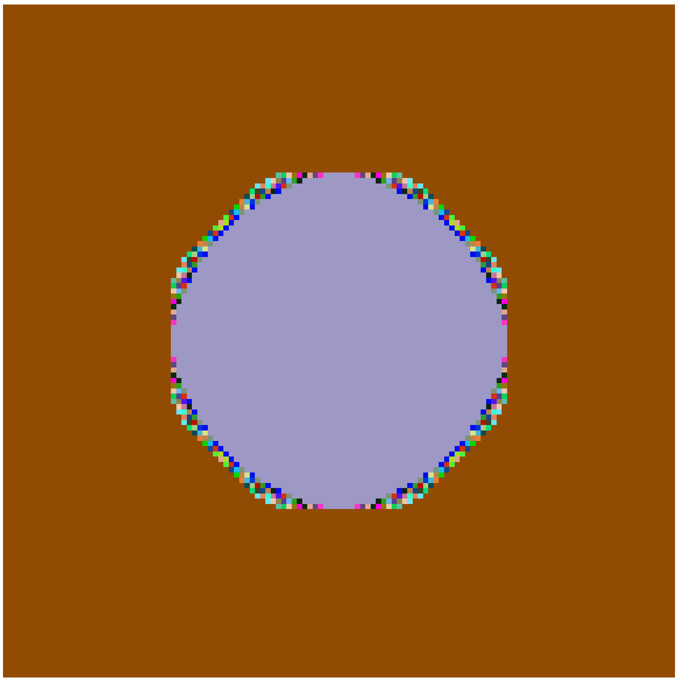

FIGURE 4. *The solution of the discrete BV problem using the upwind definition* (17) *in "false color".*

256×256, and 512×512, and measure the $L_2(I)$ distance between these approximations $\tilde{f}_h^n$ and the projection onto a 2048×2048 grid of $\tilde{f}$ given above. These differences are reported in Table 3.

We also computed solutions approximate solutions f_h^n when f is 255 times the characteristic function of the disk

$$f = 255\chi_{|x-(\frac{1}{2},\frac{1}{2})|\leq\frac{1}{4}}, \tag{31}$$

with $\|f - \tilde{f}\|_{L_2(I)} = 16, 32, 64$. Again we used Dirichlet boundary conditions and the discrete minimization problems (6) and (22); again we ensured that $\epsilon(p^n) \leq 1/4$, so we know that $\|\tilde{f}_h - \tilde{f}_h^n\|_{L_2(I)} \leq 1/4$. The exact solution $\tilde{f}$ is simply a multiple of the characteristic function of same disk such that $\|f - \tilde{f}\|_{L_2(I)}$ is the correct value. The values of $\|f - \tilde{f}\|_{L_2(I)} = 16$, 32, and 64 correspond to

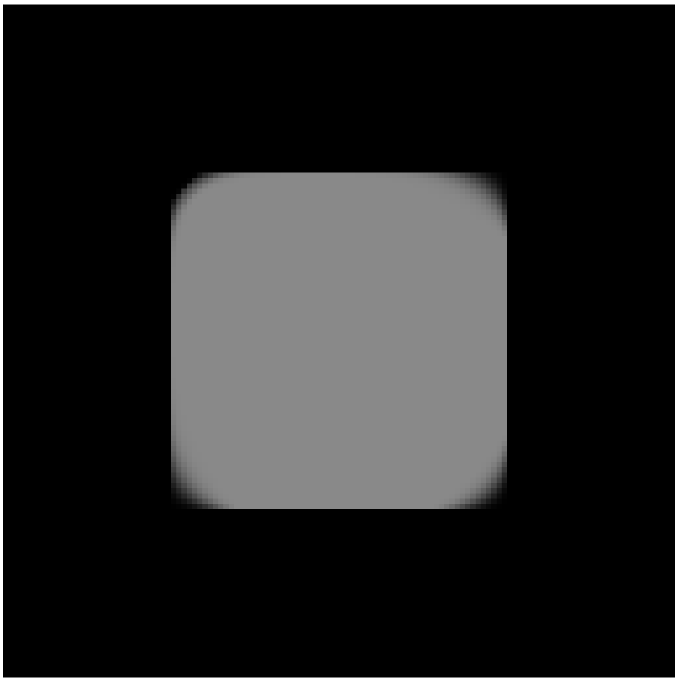

FIGURE 5. *The solution of the discrete BV problem using the anisotropic definition* (8) (128×128).

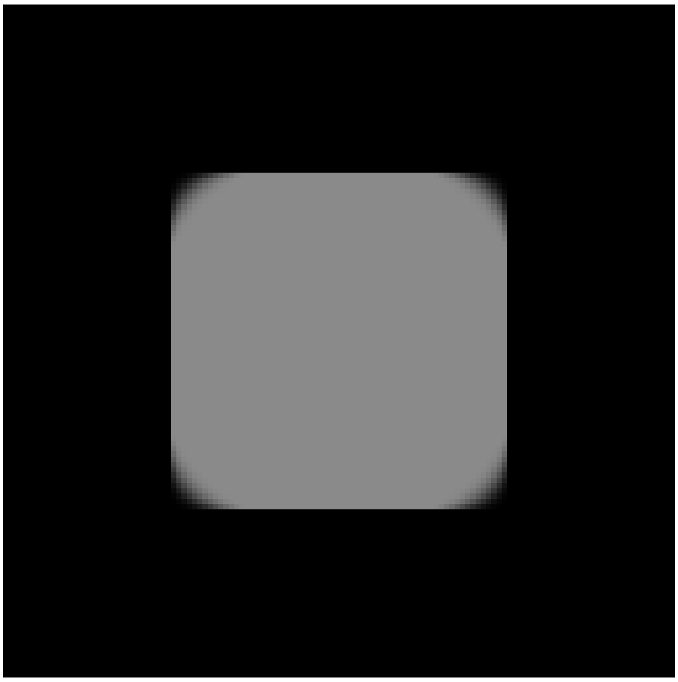

FIGURE 6. *The solution of the discrete BV problem using the upwind definition* (17) (128×128).

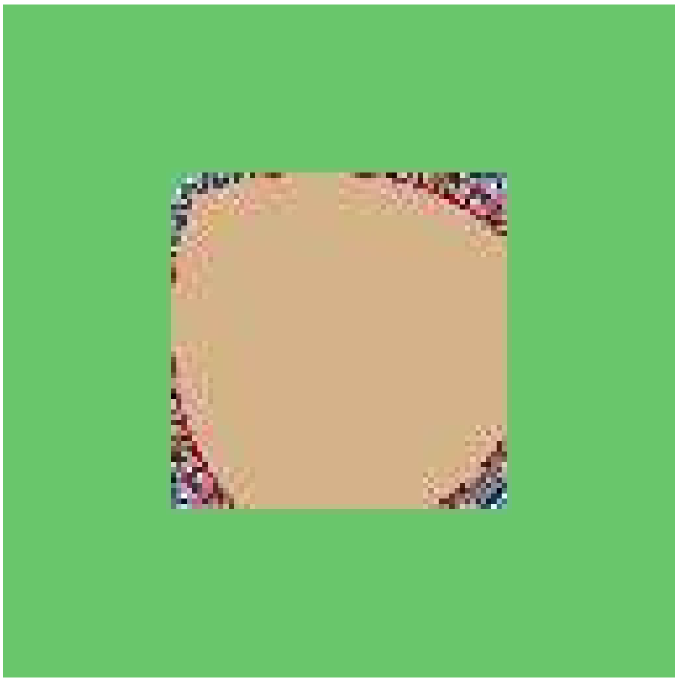

FIGURE 7. *The solution of the discrete BV problem using the anisotropic definition* (8) *in "false color"* (128×128).

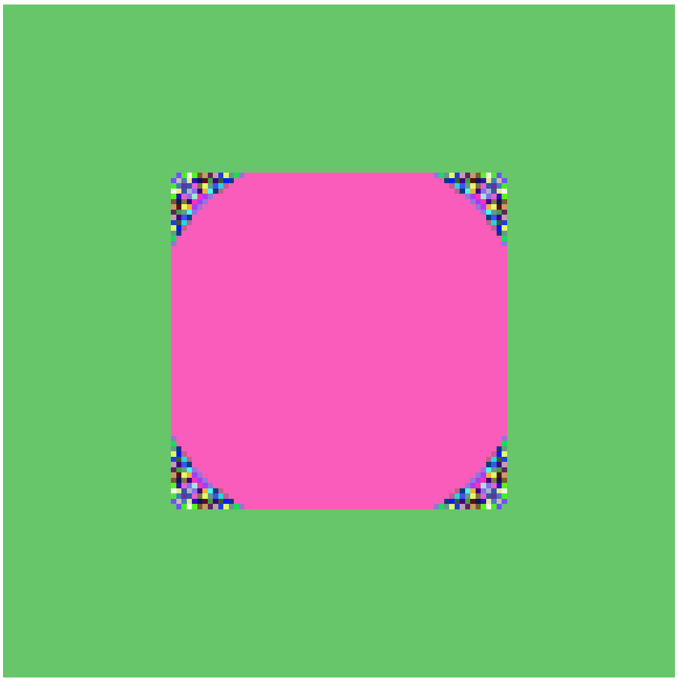

FIGURE 8. *The solution of the discrete BV problem using the upwind definition* (17) *in "false color"* (128×128).

$\lambda = 4.5134516668$, 9.02703337, and 18.05406674, respectively. We compared the same piecewise constant approximation (30) to $\tilde{f}$ on a 2048×2048 grid to piecewise constant approximations $\tilde{f}_h^n$ the discrete solutions of (6) and (22) on grids of size 128×128, 256×256, and 512×512. Here the piecewise constant, discrete initial data f_h is not equal to the true initial data f; we used the same interpolation method (30) to compute our discrete data f_h on grids of size 128×128, etc. The results are reported in Table 4.

With both the characteristic function of the disk and the square as data, the solution is in the Sobolev space $W^{\beta,2}$ for β at most $1/2$, so one might suspect that the maximum possible rate of approximation in $L_2(I)$ by piecewise constants is $O(h^\beta)$, and this is, roughly, what one observes when the data is a multiple of the characteristic function of a disk. When the data is a multiple of the characteristic function of the square, however, the discontinuities in the solution are aligned with the computational grid, and the convergence rate, which is at most one for piecewise constant approximations, is limited by the smoothness of $\tilde{f}$ inside only the subsquare $[\frac{1}{4}, \frac{3}{4}]^2$, and the experimental convergence rate is clearly above 1/2.

Next we discuss the qualitative properties of the numerical solutions.

The anisotropy of the operator (7) was briefly noted in the previous section. The operator (20) was offered as a "more isotropic" operator, but it, too, is fundamentally anisotropic—if a function has discontinuities across curves that are not vertical, horizontal, or diagonal lines then indeed

$$\lim_{h\to 0} |g|_{\mathrm{BV}^h(I)} \neq |g|_{\mathrm{BV}(I)}$$

even for the "upwind" discrete BV semi-norm. Here we briefly give two examples that illustrate the effect of this anisotropy.

As usual, we work only with Dirichlet boundary conditions. We begin with

$$f = 255\,\chi_D,$$

where D is the disk with center $\left(\frac{1}{2}, \frac{1}{2}\right)$ and radius $\frac{1}{4}$. Using the iteration (26), we then find the approximate minimizer over all discrete g with

$$\|f - g\|_{L_2^h(I)} \leq 64$$

of

$$\|g\|_{\mathrm{BV}^h(I)}$$

for both the anisotropic definition (8) and the "upwind" definition (17) of the discrete BV semi-norm. The exact minimizer of the continuous problem in this case is known [22] to be $\left(255 - \frac{2\lambda}{r}\right)\chi_D$, where $r = \frac{1}{4}$ is the radius of the disk and λ (satisfying $\lambda > \frac{1}{r}$) is chosen so that

$$\|255\,\chi_D - \left(255 - \frac{2\lambda}{r}\right)\chi_D\|_{L_2(I)} = 64.$$

We use the error bound $\epsilon(p^n)$ to ensure that the $L_2^h(I)$ errors are no greater than 1/4.

Table 4

$L_2(I)$ errors on grids of size 128, 256, and 512, and differences $\|f - \tilde{f}\|_{L_2(I)}$ of 16, 32, and 64, with initial data (31); columns 1–3 are the result of (6); columns 4–6 are the result of (22) α is the estimated order of convergence, $\|\tilde{f} - \tilde{f}_h\|_{L_2(I)} \approx Ch^{\alpha}$.

	16	32	64	16	32	64
128	10.637	9.223	6.004	9.925	8.312	5.143
256	7.929	6.981	4.542	7.061	6.051	3.795
512	6.029	5.360	3.495	5.185	4.503	2.852
α	0.410	0.392	0.390	0.468	0.442	0.425

In our experiments we set $h = 1/128$. The results are shown in Figure 1 for the anisotropic discrete BV norm (8) and in Figure 2 for the upwind discrete BV norm (17). If one looks closely, one sees that the "northeast" and "southwest" borders of the solution disk in the anisotropic solution are more smoothed than other parts of the border, and the "upwind" solution has generally a sharper border everywhere.

To illustrate this phenomenon more clearly, we include "false color" images of Figures 1 and 2 as Figures 3 and 4, respectively. Here each greyscale was assigned an arbitrary color to show how much the borders are smoothed in each of the solutions. For example, the pixels with a greyscale of 0 were colored with the terra-cotta-like color; the same mapping of greyscales to colors was used in both images.

Some things stick out immediately from the false-color images. First, the greyscale values of the central plateau of the discrete solutions are different, but their difference of only one greyscale value (113 in the anisotropic image, 112 in the greyscale image) could be explained by numerical error.

Second, the smoothing of the border of the solution truly *is* anisotropic in the "anisotropic" image, and it is smoothed over a distance of about 9 pixels in the northeast and southwest directions, which is significant (and which cannot be due to the discrete error, which as noted above is no more than 0.25 RMS greyscales). The smoothing of the upwind solution is spread over a noticeably smaller distance. The notion that "BV preserves edges", while true in the continuous setting, clearly needs some qualification in the discrete setting.

There are precisely three places where there is a one-pixel jump from the plateau of the disk to the background color in the anisotropic image—at the right, the bottom, and the "northwest" corner. In the upwind image, there are four such places, at the left and right and top and bottom edges of the plateau.

Figures 5-8 show similar qualitative effects when the initial data is the square $[\frac{1}{4}, \frac{3}{4}]^2$.

To illustrate the damage to the multiscale algorithm by (necessarily) projecting the injected vector field

$$\vec{I}^h_{2h} p_{2h}$$

onto the convex set K_h we illustrate

$$I^h_{2h} f_{2h} - \lambda \nabla_h \cdot \vec{I}^h_{2h} p_{2h} = I^h_{2h}(f_{2h} - \lambda \nabla_{2h} \cdot p_{2h}), \tag{32}$$

which is just the injection of the solution at a scale of $2h$ into the space of piecewise constants with scale h, and

$$I^h_{2h} f_{2h} - \lambda \nabla_h \cdot (\pi_{K_h} \vec{I}^h_{2h} p_{2h}), \tag{33}$$

which is close to the initial approximation to the solution at scale h. (It would be precisely the initial approximation at scale h if we had f_h instead of $I^h_{2h} f_{2h}$.) We take $h = 1/128$ and the anisotropic method (8). Figure 9 shows (32), which is the 64×64 solution injected into the 128×128 grid; Figure 10 shows (33), i.e., what happens when we project the initial vector field onto K_h. The effect is so severe

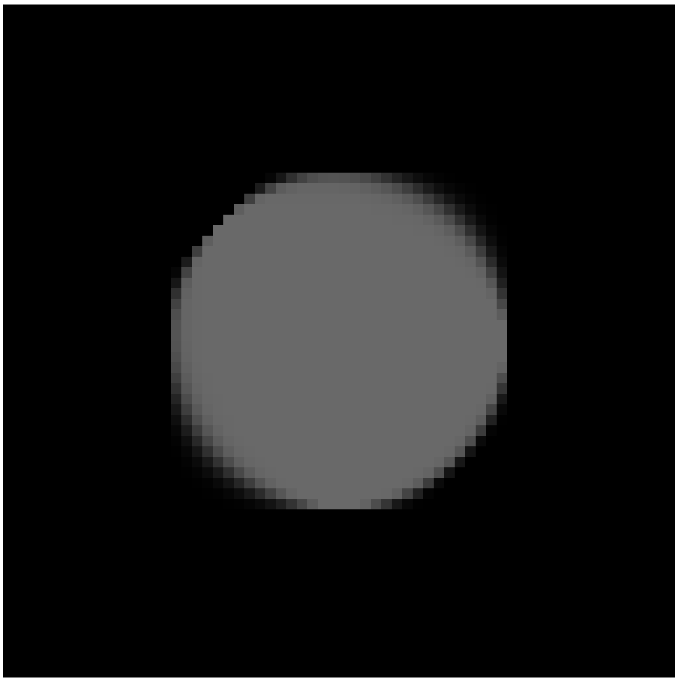

FIGURE 9. *The image* (32) *with* $h = 1/128$ *and* (8) *with* $\|\tilde{f} - f\|_{L_2(I)} = 64$.

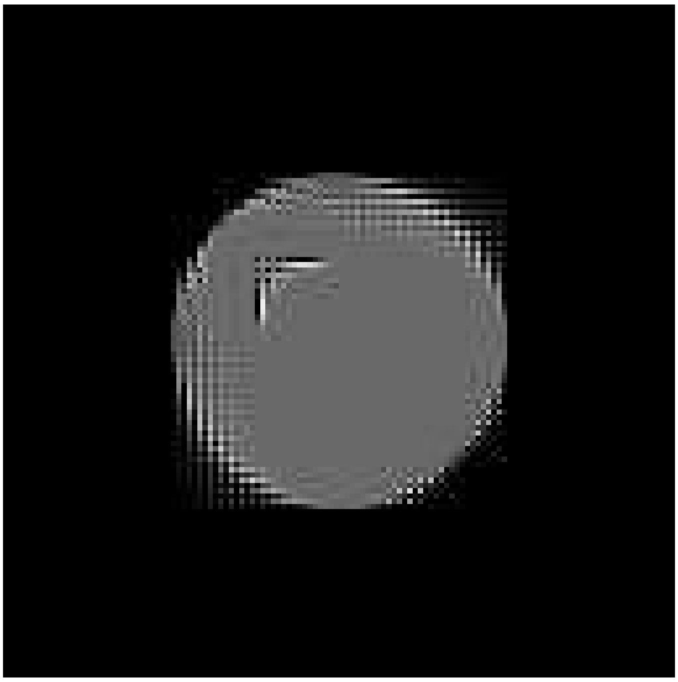

FIGURE 10. *The image* (33) *with* $h = 1/128$ *and* (8) *with* $\|\tilde{f} - f\|_{L_2(I)} = 64$.

because the operator ∇_h magnifies any changes incurred by π_{K_h} by a factor of $1/h$.

These figures strongly suggest that there should be a better way to map $\vec{I}^h_{2h} p_{2h}$ onto K_h. For example, for any divergence-free vector field q_h (with $\nabla_h \cdot q_h = 0$), we could start just as well with

$$p^0_h = \pi_{K_h}(\vec{I}^h_{2h} p_{2h} - q_h) \tag{34}$$

and still satisfy (cf. (27))

$$\nabla_h \cdot (\vec{I}^h_{2h} p_{2h} - q_h) = I^h_{2h} \nabla_{2h} \cdot p_{2h}.$$

If we could find q_h so that

$$\vec{I}^h_{2h} p_{2h} - q_h \in K_h,$$

for example, then using (34) as the initial vector field would result in Figures 9 and 10 being identical.

We now consider the example

$$f = 255\chi_S$$

where S is the square $[\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{4}, \frac{3}{4}]$. As in the last example, we found the approximate minimizer of $\|g\|_{BV}$ satisfying $\|f - g\|_{L^2_h(I)} \le 64$. Allard gives the exact minimizer of the continuous problem in an appendix to [1] (see Figure 11).

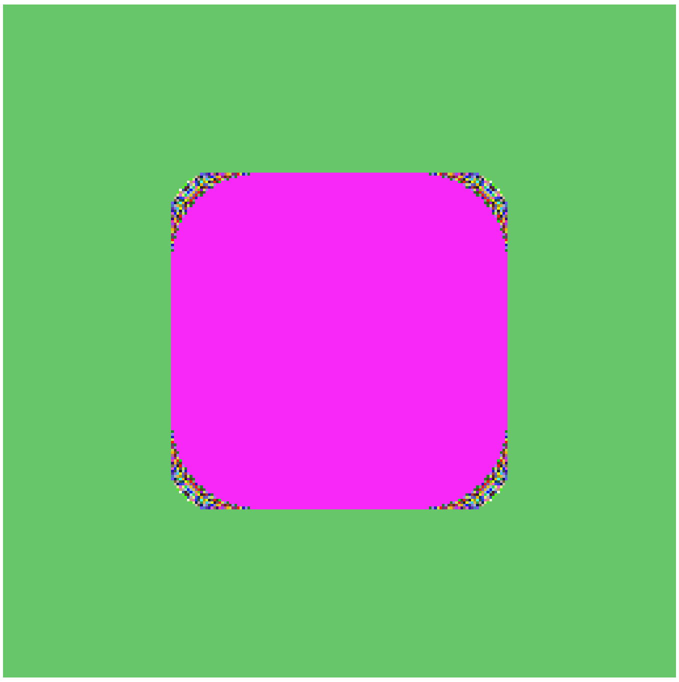

FIGURE 11. *The projection onto a* 256×256 *grid of the exact solution of the continuous BV problem, in "false color".*

Both the original and upwind solutions preserve sharp jumps at the sides of the squares, however, we see some significant differences near the corners. As expected, the upwind scheme yields the same type of smoothing at each of the four corners, which seems to match the true solution (see Figure 11). On the other hand, the anisotropy of the original scheme is evident in the behavior of the four corners. It is also curious to note that the right and bottom sides of the square are much better preserved than the top and left ones. Furthermore, the truncated top left corner appears to be favoring a sharp jump over what should be a more gradual ramp. Finally, we note the magnified anisotropy of the ramps in the northwest and southeast corners.

REFERENCES

[1] W. K. Allard, *Total variation regularization for image denoising. I. Geometric theory*, SIAM J. Math. Anal., 39 (2007/08), pp. 1150–1190.

[2] F. Alter, V. Caselles, and A. Chambolle, *Evolution of characteristic functions of convex sets in the plane by the minimizing total variation flow*, Interfaces Free Bound., 7 (2005), pp. 29–53.

[3] B. Appleton and H. Talbot, *Globally optimal geodesic active contours*, J. Math. Imaging Vision, 23 (2005), pp. 67–86.

[4] J.-F. Aujol, *Some algorithms for total variation based image restoration*, CMLA Preprint 2008-05.

[5] G. Bellettini, V. Caselles, and M. Novaga, *The total variation flow in* $\mathbb{R}^N$, J. Differential Equations, 184 (2002), pp. 475–525.

[6] A. Braides, *Γ-convergence for beginners*, Oxfor Lecture Series in Mathematics and it Applications, Vol. 22, Oxford University Press, Oxford, UK, 2002.

[7] X. Bresson and T. F. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems and Imaging, 2 (2008), pp. 455–484.

[8] M. Burger, G. Gilboa, S. Osher, and J. Xu, *Nonlinear inverse scale space methods*, Commun. Math. Sci., 4 (2006), pp. 179–212.

[9] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[10] ———, *Total variation minimization and a class of binary MRF models*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Proceedings of the Fifth International Workshop, EMMCVPR 2005, St. Augustine, FL, USA, November 9-11, 2005 LNCS 3757, Anand Rangarajan, Baba Vemuri, Alan L. Yuille, eds., Springer-Verlag, Berlin, 2005, pp. 136–152.

[11] A. Chambolle and P.-L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.

[12] P. L. Combettes, *Solving monotone inclusions via compositions of nonexpansive averaged operators*, Optimization, 53 (2004), pp. 475–504.

[13] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200.

[14] J. Darbon and M. Sigelle, *Exact optimization of discrete constrained total variation minimization problems*, in Combinatorial Image Analysis, Lecture Notes in Comput. Sci. 3322, Springer-Verlag, Berlin, 2004, pp. 548–557.

[15] R. DeVore, B. Jawerth, and B. Lucier, *Image Compression through wavelet transform coding*, IEEE Trans. Information Theory, 38, 2 (1992), pp. 719–746; Special issue on Wavelet Transforms and Multiresolution Analysis.

[16] B. Eicke, *Iteration methods for convexly constrained ill-posed problems in Hilbert space*, Numer. Funct. Anal. Optim, 13 (1992), pp. 413–429.

[17] I. Ekeland and R. Temam, *Convex analysis and variational problems*, Translated from the French, Studies in Mathematics and it's Applications, Vol. 1, North-Holland Publishing Co., Amsterdam, 1976.

[18] D. Goldfarb and W. Yin, *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput., 27 (2005), pp. 622–645.

[19] Z. Opial, *Weak convergence of the sequence of successive approximations for nonexpansive mappings*, Bull. Amer. Math. Soc., 73 (1967), pp. 591–597.

[20] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.

[21] S. Osher and J. S. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys., 79 (1988), pp. 12–49.

[22] W. Ring, *Structural properties of solutions to total variation regularization problems*, M2AN Math. Model. Numer. Anal., 34 (2000), pp. 799–810.

[23] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.

[24] J. Wang and B. J. Lucier, *Error bounds for finite difference methods for ROF image smoothing*, to appear.